Homework 10

Christophe Hunt

April 3, 2017

Contents

1 1. Problem Set

Playing with PageRank You'll verify for yourself that PageRank works by performing calculations on a small universe of web pages. Let's use the 6 page universe that we had in the course notes.

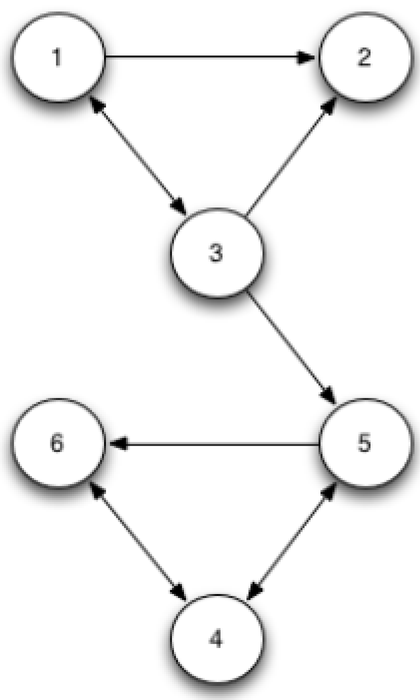


FIGURE 1. Toy Example with 6 URLs.

Figure 1:

For this directed graph, perform the following calculations in R.

1.1 1

Form the A matrix. Then, introduce decay and form the B matrix as we did in the course notes.

$$\mathbf{A} = \begin{bmatrix} 0 & \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{3} & 0 & 0 & \frac{1}{3} & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Figure 2:

```
A <- matrix(data = c(0,0.5,0.5,0,0,0,
                     0,0,0,0,0,0,
                     0.33,0.33,0,0,0.33,0,
                     0,0,0,0,0.5,0.5,
                     0,0,0,0.5,0,0.5,
                     0,0,0,1,0,0), nrow = 6,  byrow = TRUE)
A
```

0.00	0.50	0.5	0.0	0.00	0.0
0.00	0.00	0.0	0.0	0.00	0.0
0.33	0.33	0.0	0.0	0.33	0.0
0.00	0.00	0.0	0.0	0.50	0.5
0.00	0.00	0.0	0.5	0.00	0.5
0.00	0.00	0.0	1.0	0.00	0.0

Now we introduce decay to create the B matrix using the equation: $B = 0.85 * A + \frac{0.15}{n}$

```
B <- 0.85 * A + .15 / nrow(A)
B
```

0.0250	0.4500	0.450	0.025	0.0250	0.025
0.0250	0.0250	0.025	0.025	0.0250	0.025
0.3055	0.3055	0.025	0.025	0.3055	0.025
0.0250	0.0250	0.025	0.025	0.4500	0.450
0.0250	0.0250	0.025	0.450	0.0250	0.450
0.0250	0.0250	0.025	0.875	0.0250	0.025

1.2 2

Start with a uniform rank vector r and perform power iterations on B till convergence. That is, compute the solution $r = B^n * r$. Attempt this for a sufficiently large n so that r actually converges.

Following our notes, we assume equal probability of a user clicking a link so we start with the rank vector r is $r = (.167, .167, .167, .167, .167, .167)$

```
library(expm)
```

```
## Warning: package 'expm' was built under R version 3.3.3
```

```
## Loading required package: Matrix

##
## Attaching package: 'expm'

## The following object is masked from 'package:Matrix':
##
##     expm
```

```
r <- c(.167, .167, .167, .167, .167, .167)
n <- 40
(B %^% n) %*% r
```

0.0100757
0.0026836
0.0138653
0.0251686
0.0251686
0.0251686

```
A %*% r
```

0.16700
0.00000
0.16533
0.16700
0.16700
0.16700

1.3 3

Compute the eigen-decomposition of B and verify that you indeed get an eigenvalue of 1 as the largest eigenvalue and that its corresponding eigenvector is the same vector that you obtained in the previous power iteration method. Further, this eigenvector has all positive entries and it sums to 1.

1.4 4

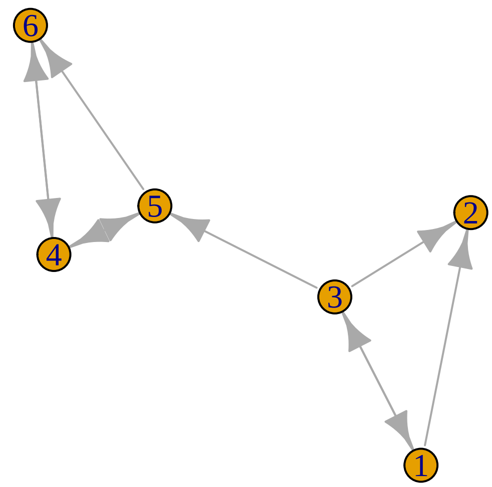
```r
library(igraph)
```

Use the $graph$ package in R and its $page.rank$ method to compute the Page Rank of the graph as given in A. Note that you don't need to apply decay. The package starts with a connected graph and applies decay internally. Verify that you do get the same PageRank vector as the two approaches above. Please document all your experiments in an R Markdown document.

```r
library(igraph)
df <- data.frame(from=c(1,1,3,3,3,4,4,5,5,6),
                  to = c(2,3,1,2,5,5,6,4,6,4),
              weight = c(1,1,1,1,1,1,1,1,1,1))
g <- graph.data.frame(df, directed = TRUE)
results <- page_rank(g, algo = c("power"))
v <- matrix(results$vector, ncol = 1)
v
```

0.0395097
0.0440479
0.3705699
0.2052394
0.2833665
0.0572665

```r
igraph::plot.igraph(g)
```


6
5
4
3
2
1